problema ***Mappa antica (mappa)*** caso di test ***4x4*** https://training.olinfo.it/#/task/mappa/statement

La soluzione proposta nella *Guida per le selezioni territoriali* del prof. Bugatti - sesta edizione (pag. 58) utilizza la funzione ricorsiva **attraversa**:

```
#include <iostream>
#include <fstream>
using namespace std;
#include <algorithm>
#define DIM 100

  int mappa[100][100];
  int N;

void attraversa (int i , int j){
      for (int r=-1; r<2; r++)    // -1, 0, 1 precedente attuale successivo
          for (int s=-1; s<2; s++){
              if (i+r < 0 || i+r > N-1
              || j+s < 0 || j+s > N-1);              //--- if1 ---
//---------- allora ---------- NESSUNA AZIONE (fuori mappa)
              else                                   //--- if2---
                if (mappa[i+r][j+s] == 0
                || mappa[i+r][j+s] > mappa[i][j] + 1 ){
                    mappa[ i+r ] [ j+s ] = mappa[ i ] [ j ] + 1;
                  attraversa (i+r , j+s);
                }
//---------- altrimenti NESSUNA AZIONE (trabocchetto -1/+ oppure NON maggiore)
            }
}
```

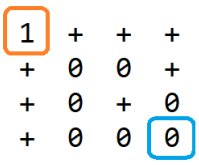
```
int main() {
  ifstream in ("input.txt");
  ofstream out("output.txt");

   in >> N;
   char c;

   for (int i=0;i < N;i++)
       for (int j=0;j <N ;j++){
           in >> c;
           if (c=='*')
              mappa[i][j]= 0;
           else
              mappa[i][j]= -1;
   }

       mappa[0][0] = 1;
       attraversa (0 , 0);
       out << mappa[N-1][N-1];

       return 0;
}
```

File input.txt	File output.txt
4 *+++ +**+ +*+* +***	5

Il file di input proposto nella *Guida* è un **labirinto a scacchiera 4x4** →

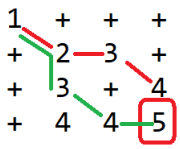
In questo caso la soluzione è **5**, il minor numero possibile di lastroni innocui da calpestare per raggiungere la posizione (3,3) partendo da (0,0)

Ogni casella della tabella `mappa` indicherà, a fine programma, la distanza minima di quella casella dalla casella di partenza oppure un trabocchetto mortale rappresentato con il valore -1. Rappresentando in forma mista i valori della tabella (con + invece che con -1 i trabocchetti e con un valore numerico le distanze), **al primo richiamo** della funzione ricorsiva la tabella `mappa` conterrà:



← **1** nella casella di partenza di coordinate (0,0) e 0 nelle caselle senza trabocchetto, compresa la casella di destinazione (3,3)

quando tutte le caselle saranno state visitate, nella casella nell'angolo in basso a destra (3,3) ci sarà la lunghezza del percorso migliore: 5 →



A partire dalla casella (0,0) si visitano in maniera ricorsiva le 8 caselle vicine (quadrato 3x3 di **centro (i,j)** delimitato dal **bordo rosso** nelle immagini che seguono). **Se la casella visitata è interna alla tabella `mappa`** e il valore nella casella visitata è maggiore del valore della casella più 1 oppure se è 0 (cioè non è stata ancora visitata) si aggiorna il valore della casella visitata (evidenziata in verde) inserendo il valore della cella centrale più uno: **mappa[i] [j] + 1**.

Al primo richiamo della funzione ricorsiva - attraversa (0,0) - il centro del quadrato 3x3 è la casella **(0,0)** che è stata inizializzata a **1** e le celle della riga superiore (-1, -1) e (-1,0) (-1, 1) e della colonna a sinistra (0, -1) e (1,-1) sono **posizioni esterne** alla tabella **`mappa`** per le quali la funzione non esegue NESSUNA AZIONE (nell' if1 è VERA almeno una delle condizioni 1-2-3 oppure 4). Per le celle che contengono **-1** (rappresentato graficamente con **+**) le condizioni 1 e 2 dell' **if2** sono sempre **FALSE**; esempio: le posizioni (0,1) e (1,0) contengono **-1** quindi la funzione non esegue **NESSUNA AZIONE**.

La cella (1,1) è la prima cella che soddisfa la condizione 1 dell' **if2** (`mappa[i+r][j+s] == 0`), quindi viene aggiornato il valore di **mappa[1] [1]** con **mappa[0] [0] + 1** cioè con **2** (=1+1); quindi la prima casella da modificare è la (1,1) - angolo destro in basso del quadrato rosso - che diventa il centro considerato dal successivo richiamo della funzione - attraversa (1,1).

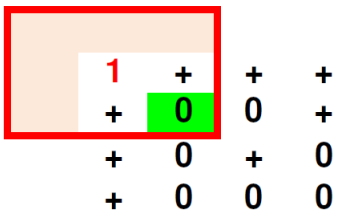
Al secondo richiamo il centro del quadrato 3x3 è la casella **(1,1)** che ora vale **2** e la prima casella da modificare è la (1,2) che diventa il centro considerato dal successivo richiamo della funzione - attraversa (1,2).

problema ***Mappa antica (mappa)*** caso di test ***4x4***

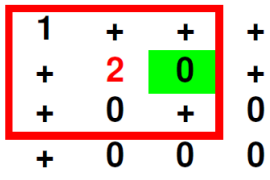
Si prosegue così fino al completamento di un livello della funzione ricorsiva (in questo esempio fino al completamento di <7> attraversa(3,3); che restituisce il controllo al chiamante cioè a <6> attraversa(2,3);

Si riporta di seguito la sequenza degli stati della tabella `mappa` ai successivi richiami della funzione ricorsiva :

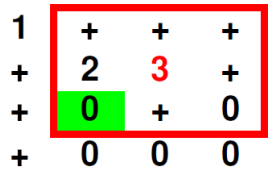
<0> attraversa(0,0)



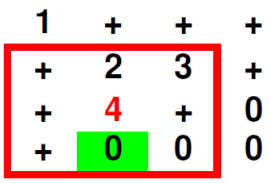
<1> attraversa(1,1)



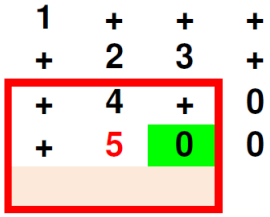
<2> attraversa(1,2)



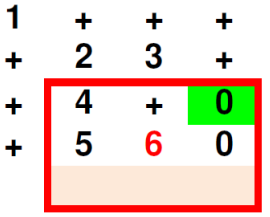
<3> attraversa(2,1)



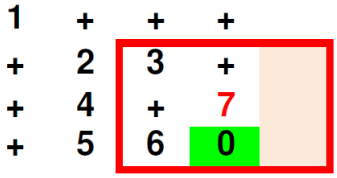
<4> attraversa(3,1)



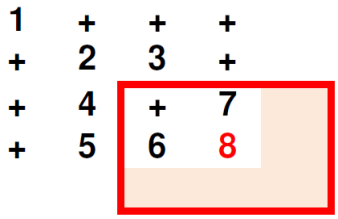
<5> attraversa(3,2)



<6> attraversa(2,3)



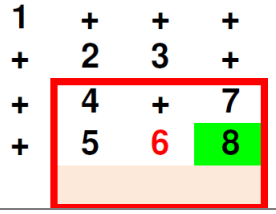
<7> attraversa(3,3) ---END



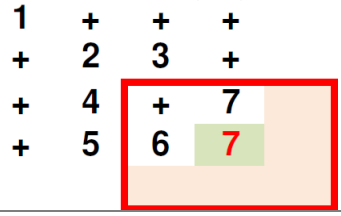
POI

<6> attraversa(2,3) ---END

<5> attraversa(3,2)



<8> attraversa(3,3) ---END

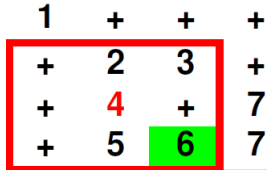


POI

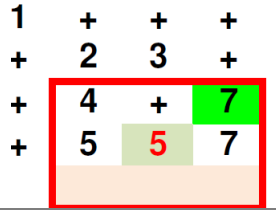
<5> attraversa(3,2) ---END

<4> attraversa(3,1) ---END

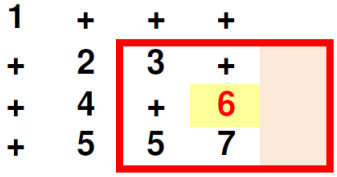
<3> attraversa(2,1)



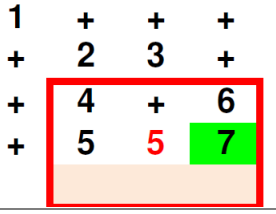
<9> attraversa(3,2)



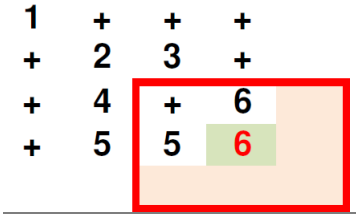
<10> attraversa(2,3) ---END



<9> attraversa(3,2)



<11> attraversa(3,3) ---END

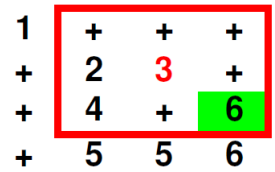


POI

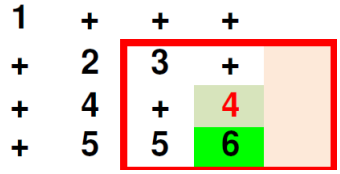
<9> attraversa(3,2) ---END

<3> attraversa(2,1) ---END

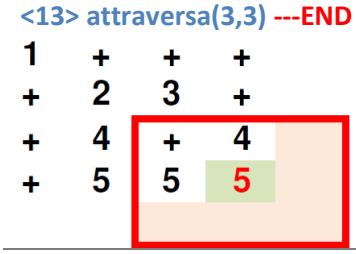
<2> attraversa(1,2)



<12> attraversa(2,3)



<13> attraversa(3,3) ---END



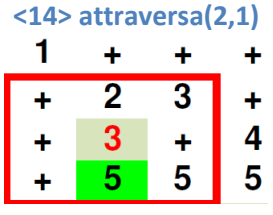
POI

<12> attraversa(2,3) ---END

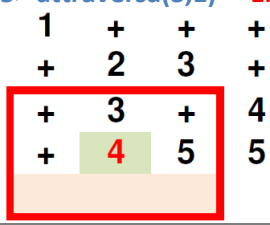
<2> attraversa(1,2) ---END

<1> attraversa(1,1)

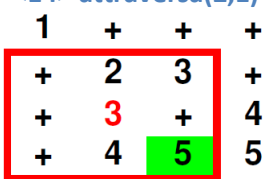
<14> attraversa(2,1)



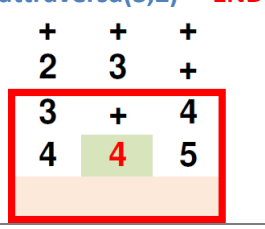
<15> attraversa(3,1) ---END



<14> attraversa(2,1)



<16> attraversa(3,2) ---END



POI

<14> attraversa(2,1) ---END

<1> attraversa(1,1) ---END

<0> attraversa(0,0) ---END